Red Hat Enterprise Linux 5 Global Network Block Device

Using GNBD with Red Hat Global File System
Edition 3

Landmann

Red Hat Enterprise Linux 5 Global Network Block Device

Using GNBD with Red Hat Global File System
Edition 3

Landmann
rlandmann@redhat.com

Legal Notice

Copyright © 2011 Red Hat, Inc.

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This book provides an overview on using Global Network Block Device (GNBD) with Red Hat GFS for Red Hat Enterprise Linux 5.

Table of Contents

Introduction **3**
1. About This Guide 3
2. Audience 3
3. Software Versions 3
4. Related Documentation 3
5. Feedback 4
6. Document Conventions 4
6.1. Typographic Conventions 4
6.2. Pull-quote Conventions 6
6.3. Notes and Warnings 6

Chapter 1. Using GNBD with Red Hat GFS **8**

Chapter 2. Considerations for Using GNBD with Device-Mapper Multipath **9**
2.1. Linux Page Caching 9
2.2. Fencing GNBD Server Nodes 9

Chapter 3. GNBD Driver and Command Usage **10**
3.1. Exporting a GNBD from a Server 10

Usage **10**

Examples **12**
3.2. Importing a GNBD on a Client 12

Usage **12**

Example **12**

Chapter 4. Running GFS on a GNBD Server Node **13**

Revision History **14**

Index **14**
D 14
E 14
F 14
G 14
I 14
L 15
S 15

Introduction

1. About This Guide

This book describes how to use Global Network Block Device (GNDB) with Global File System (GFS), including information about device-mapper multipath, GNDB driver and command usage, and running GFS on a GNBD server node.

2. Audience

This book is intended to be used by system administrators managing systems running the Linux operating system. It requires familiarity with Red Hat Enterprise Linux 5 and GFS file system administration.

3. Software Versions

Table 1. Software Versions

Software	Description
RHEL5	refers to RHEL5 and higher
GFS	refers to GFS for RHEL5 and higher

4. Related Documentation

For more information about using Red Hat Enterprise Linux, refer to the following resources:

- *Red Hat Enterprise Linux Installation Guide* — Provides information regarding installation of Red Hat Enterprise Linux 5.
- *Red Hat Enterprise Linux Deployment Guide* — Provides information regarding the deployment, configuration and administration of Red Hat Enterprise Linux 5.

For more information about Red Hat Cluster Suite for Red Hat Enterprise Linux 5, refer to the following resources:

- *Red Hat Cluster Suite Overview* — Provides a high level overview of the Red Hat Cluster Suite.
- *Configuring and Managing a Red Hat Cluster* — Provides information about installing, configuring and managing Red Hat Cluster components.
- *Logical Volume Manager Administration* — Provides a description of the Logical Volume Manager (LVM), including information on running LVM in a clustered environment.
- *Global File System: Configuration and Administration* — Provides information about installing, configuring, and maintaining Red Hat GFS (Red Hat Global File System).
- *Global File System 2: Configuration and Administration* — Provides information about installing, configuring, and maintaining Red Hat GFS2 (Red Hat Global File System 2).
- *Using Device-Mapper Multipath* — Provides information about using the Device-Mapper Multipath feature of Red Hat Enterprise Linux 5.
- *Linux Virtual Server Administration* — Provides information on configuring high-performance systems and services with the Linux Virtual Server (LVS).
- *Red Hat Cluster Suite Release Notes* — Provides information about the current release of Red Hat

Cluster Suite.

Red Hat Cluster Suite documentation and other Red Hat documents are available in HTML, PDF, and RPM versions on the Red Hat Enterprise Linux Documentation CD and online at http://www.redhat.com/docs/.

5. Feedback

If you spot a typo, or if you have thought of a way to make this manual better, we would love to hear from you. Please submit a report in Bugzilla (http://bugzilla.redhat.com/bugzilla/) against the component **rh-cs**.

Be sure to mention the manual's identifier:

```
Bugzilla component: Documentation-cluster
Book identifier: Global_Network_Block_Device(EN)-5 (2011-07-21T10:07)
```

By mentioning this manual's identifier, we know exactly which version of the guide you have.

If you have a suggestion for improving the documentation, try to be as specific as possible. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

6. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the Liberation Fonts set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later include the Liberation Fonts set by default.

6.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

Mono-spaced Bold

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

> To see the contents of the file **my_next_bestselling_novel** in your current working directory, enter the **cat my_next_bestselling_novel** command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

> Press **Enter** to execute the command.

Press **Ctrl**+**Alt**+**F2** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **mono-spaced bold**. For example:

File-related classes include **filesystem** for file systems, **file** for files, and **dir** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, select the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

6.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                  struct kvm_assigned_pci_dev *assigned_dev)
{
         int r = 0;
         struct kvm_assigned_dev_kernel *match;

         mutex_lock(&kvm->lock);

         match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                       assigned_dev->assigned_dev_id);
         if (!match) {
                 printk(KERN_INFO "%s: device hasn't been assigned before, "
                   "so cannot be deassigned\n", __func__);
                 r = -EINVAL;
                 goto out;
         }

         kvm_deassign_device(kvm, match);

         kvm_free_assigned_device(kvm, match);

out:
         mutex_unlock(&kvm->lock);
         return r;
}
```

6.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

Chapter 1. Using GNBD with Red Hat GFS

GNBD (Global Network Block Device) provides block-level storage access over an Ethernet LAN. GNBD components run as a client in a GFS node and as a server in a GNBD server node. A GNBD server node exports block-level storage from its local storage (either directly attached storage or SAN storage) to a GFS node.

Table 1.1, “GNBD Software Subsystem Components” summarizes the GNBD software subsystems components.

Table 1.1. GNBD Software Subsystem Components

Software Subsystem	Components	Description
GNBD	**gnbd.ko**	Kernel module that implements the GNBD device driver on clients.
	gnbd_export	Command to create, export and manage GNBDs on a GNBD server.
	gnbd_import	Command to import and manage GNBDs on a GNBD client.
	gnbd_serv	A server daemon that allows a node to export local storage over the network.

You can configure GNBD servers to work with device-mapper multipath. GNBD with device-mapper multipath allows you to configure multiple GNBD server nodes to provide redundant paths to the storage devices. The GNBD servers, in turn, present multiple storage paths to GFS nodes via redundant GNBDs. When using GNBD with device-mapper multipath, if a GNBD server node becomes unavailable, another GNBD server node can provide GFS nodes with access to storage devices.

This document how to use GNBD with Red Hat GFS and consists of the following chapters:

- Chapter 2, *Considerations for Using GNBD with Device-Mapper Multipath*, which describes some of the issues you should take into account when configuring multipathed GNBD server nodes
- Chapter 3, *GNBD Driver and Command Usage*, which describes the restrictions that apply when you are running GFS on a GNBD server node
- Chapter 4, *Running GFS on a GNBD Server Node*, which describes the user commands that configure GNBD

Chapter 2. Considerations for Using GNBD with Device-Mapper Multipath

GNBD with device-mapper multipath allows you to configure multiple GNBD server nodes (nodes that export GNBDs to GFS nodes) to provide redundant paths to the storage devices. The GNBD server nodes, in turn, present multiple storage paths to GFS nodes via redundant GNBDs. When using GNBD with device-mapper multipath, if a GNBD server node becomes unavailable, another GNBD server node can provide GFS nodes with access to storage devices.

If you are using GNBD with device-mapper multipath, you need to take the following into consideration:

- Linux page caching, as described in Section 2.1, “Linux Page Caching”.
- Fencing GNBD server nodes, as described in Section 2.2, “Fencing GNBD Server Nodes”.
- GNBD device names; export names for GNBD devices must be unique. Additionally, you must specify the **-u** or **-U** when using the **gnbd_export** command. Exporting GNBD devices is described in Chapter 3, *GNBD Driver and Command Usage*.

2.1. Linux Page Caching

For GNBD with device-mapper multipath, do *not* specify Linux page caching (the **-c** option of the **gnbd_export** command). All GNBDs that are part of a logical volume must run with caching *disabled*. Data corruption occurs if the GNBDs are run with caching enabled. Refer to Section 3.1, “Exporting a GNBD from a Server” for more information about using the **gnbd_export** command for GNBD with device-mapper multipath.

2.2. Fencing GNBD Server Nodes

GNBD server nodes must be fenced using a fencing method that physically removes the nodes from the network. To physically remove a GNBD server node, you can use any fencing device: *except* the following: **fence_brocade** fence agent, **fence_vixel** fence agent, **fence_mcdata** fence agent, **fence_sanbox2** fence agent, **fence_scsi** fence agent. In addition, you *cannot* use the GNBD fencing device (**fence_gnbd** fence agent) to fence a GNBD server node. For information about configuring fencing for GNBD server nodes, refer to the *Global File System* manual.

Chapter 3. GNBD Driver and Command Usage

The Global Network Block Device (GNBD) driver allows a node to export its local storage as a GNBD over a network so that other nodes on the network can share the storage. Client nodes importing the GNBD use it like any other block device. Importing a GNBD on multiple clients forms a shared storage configuration through which GFS can be used.

The GNBD driver is implemented through the following components.

- **gnbd_serv** — Implements the GNBD server. It is a user-space daemon that allows a node to export local storage over a network.
- **gnbd.ko** — Implements the GNBD device driver on GNBD clients (nodes using GNBD devices).

Two user commands are available to configure GNBD:

- **gnbd_export** (for servers) — User program for creating, exporting, and managing GNBDs on a GNBD server.
- **gnbd_import** (for clients) — User program for importing and managing GNBDs on a GNBD client.

3.1. Exporting a GNBD from a Server

The **gnbd_serv** daemon must be running on a node before it can export storage as a GNBD. You can start the **gnbd_serv** daemon running **gnbd_serv** as follows:

```
#gnbd_serv
gnbd_serv: startup succeeded
```

Once local storage has been identified to be exported, the **gnbd_export** command is used to export it.

Note

When you configure GNBD servers with device-mapper multipath, you must not use page caching. All GNBDs that are part of a logical volume must run with caching disabled. By default, the **gnbd_export** command exports with caching turned off.

Note

A server should not import the GNBDs to use them as a client would. If a server exports the devices uncached, the underlying devices may also be used by **gfs**.

Usage

```
gnbd_export -d pathname -e gnbdname [-c][-u][-U
```

pathname

Specifies a storage device to export.

gnbdname

Specifies an arbitrary name selected for the GNBD. It is used as the device name on GNBD clients. This name must be unique among all GNBDs exported in a network.

-o

Export the device as read-only.

-c

Enable caching. Reads from the exported GNBD and takes advantage of the Linux page cache.

By default, the **gnbd_export** command does *not* enable caching.

Note

When you configure GNBD servers with device-mapper multipath, do not specify the **-c** option. All GNBDs that are part of a logical volume must run with caching disabled.

Note

If you have been using GFS 5.2 or earlier and do *not* want to change your GNBD setup you *should* specify the **-c** option. Before GFS Release 5.2.1, Linux caching was enabled by default for **gnbd_export**. If the **-c** option is *not* specified, GNBD runs with a noticeable performance decrease. Also, if the **-c** option is *not* specified, the exported GNBD runs in timeout mode, using the default timeout value (the **-t** option). For more information about the **gnbd_export** command and its options, refer to the **gnbd_export** man page.

-u *uid*

Manually sets the Universal Identifier for an exported device. This option is used with **-e**. The UID is used by device-mapper multipath to determine which devices belong in a multipath map. A device must have a UID to be multipathed. However, for most SCSI devices the default Get UID command, **/usr/sbin/gnbd_get_uid**, will return an appropriate value.

Note

The UID refers to the device being exported, not the GNBD itself. The UIDs of two GNBD devices should be equal, only if they are exporting the same underlying device. This means that both GNBD servers are connected to the same physical device.

-U *Command*

Gets the UID command. The UID command is a command the **gnbd_export** command will run to get a Universal Identifier for the exported device. The UID is necessary to use device-mapper multipath with GNBD. The command must use the full path of any executeable that you wish to run. A command can contain the %M, %m or %n escape sequences. %M will be expanded to the major number of the exported device, %m will be expanded to the minor number of the exported device, and %n will be expanded to the **sysfs** name for the device. If no command is given, GNBD will use the default command **/usr/sbin/gnbd_get_uid**. This command will work for most SCSI devices.

Examples

This example is for a GNBD server configured with GNBD multipath. It exports device **/dev/sdc2** as GNBD **gamma**. Cache is disabled by default.

```
gnbd_export -d /dev/sdc2 -e gamma -U
```

This example is for a GNBD server *not* configured with GNBD multipath. It exports device **/dev/sdb2** as GNBD **delta** with cache enabled.

```
gnbd_export -d /dev/sdb1 -e delta -c
```

This example exports device **/dev/sdb2** as GNBD **delta** with cache enabled.

```
gnbd_export -d /dev/sdb2 -e delta -c
```

3.2. Importing a GNBD on a Client

The **gnbd.ko** kernel module must be loaded on a node before it can import GNBDs. When GNBDs are imported, device nodes are created for them in **/dev/gnbd/** with the name assigned when they were exported.

Usage

```
gnbd_import -i Server
```

Server

Specifies a GNBD server by hostname or IP address from which to import GNBDs. All GNBDs exported from the server are imported on the client running this command.

Example

This example imports all GNBDs from the server named **nodeA**.

```
gnbd_import -i nodeA
```

Chapter 4. Running GFS on a GNBD Server Node

You can run GFS on a GNBD server node, with some restrictions. In addition, running GFS on a GNBD server node reduces performance. The following restrictions apply when running GFS on a GNBD server node.

Important

When running GFS on a GNBD server node you *must* follow the restrictions listed; otherwise, the GNBD server node will fail.

1. A GNBD server node must have local access to all storage devices needed to mount a GFS file system. The GNBD server node must not import (**`gnbd_import`** command) other GNBD devices to run the file system.
2. The GNBD server must export all the GNBDs in uncached mode, and it must export the raw devices, not logical volume devices.
3. GFS must be run on top of a logical volume device, not raw devices.

Note

You may need to increase the timeout period on the exported GNBDs to accommodate reduced performance. The need to increase the timeout period depends on the quality of the hardware.

Revision History

Revision 3-11.400	**2013-10-31**	**Rüdiger Landmann**
Rebuild with publican 4.0.0		
Revision 3-11	**2012-07-18**	**Anthony Towns**
Rebuild for Publican 3.0		
Revision 5.7-1	**Thu Jul 21 2011**	**Steven Levine**
Resolves: #703610 Fixes minor typographical errors.		
Revision 1.0-0	**Thu Jan 29 2009**	

Index

D

device-mapper multipath, Considerations for Using GNBD with Device-Mapper Multipath
- fencing GNBD server nodes, Fencing GNBD Server Nodes
- Linux page caching, Linux Page Caching

driver and command usage, GNBD Driver and Command Usage
- exporting from a server, Exporting a GNBD from a Server
- importing on a client, Importing a GNBD on a Client

E

exporting from a server daemon, Exporting a GNBD from a Server

F

feedback, Feedback

fencing GNBD server nodes, Fencing GNBD Server Nodes

G

GFS, using on a GNBD server node, Running GFS on a GNBD Server Node

GNBD, using with Red Hat GFS, Using GNBD with Red Hat GFS

gnbd.ko module, GNBD Driver and Command Usage, Importing a GNBD on a Client

gnbd_export command , GNBD Driver and Command Usage, Usage

gnbd_import command , GNBD Driver and Command Usage, Usage

gnbd_serv daemon, GNBD Driver and Command Usage, Exporting a GNBD from a Server

I

importing on a client module, Importing a GNBD on a Client

L

Linux page caching, Linux Page Caching

S

software subsystem components, Using GNBD with Red Hat GFS